Physical Layer Secret-Key Generation with Discreet Cosine Transform for the Internet of Things

George Margelis*, Xenofon Fafoutis*, George Oikonomou*†, Robert Piechocki*, Theo Tryfonas†, Paul Thomas‡
*Communication Systems and Networks Research Group, University of Bristol, Bristol, UK
†Bristol Cryptography Group, University of Bristol, Bristol, UK
‡University of Bristol Honorary Research Fellow
george.margelis@bristol.ac.uk

***Abstract*—The confidentiality of communications in the Internet of Things (IoT) is critical, with cryptography currently being the most widely employed method of ensuring it. Establishing cryptographically secure communication links between two transceivers requires the pre-agreement on some key, unknown to an external attacker. In recent years there has been growing attention in techniques that generate a shared random key through observation of the channel and its effects on the exchanged messages. In this work we present SKYGlow, a novel scheme for secret-key generation, designed for IoT devices, such as IEEE 802.15.4 and Bluetooth Low Energy (BLE) transceivers. SKYGlow employs the Discreet Cosine Transform (DCT) of channel observations and Slepian-Wolf coding for information reconciliation. Real-life experiments have resulted in the creation of 128-bit secret keys with only 65 packet exchanges and with an entropy of 0.9978 bits, making our scheme much more energy-efficient compared with others in the existing literature.**

I. Introduction

The vision of an Internet of Things (IoT) is coming closer to realization with each passing day, where physical objects will have virtual representations, and the ability to be remotely controlled or act as physical access points to Internet services [1]. However, this vision introduces new security risks since attackers can potentially gain access to systems considered so far as secure. Furthermore, the broadcasting nature of Wireless Sensor Networks (WSN), which will form a large part of the IoT, makes communications prone to eavesdropping, increasing the need for confidentiality, which currently is accomplished by cryptographic schemes.

Unfortunately, the nodes that will comprise these WSN are very constrained in hardware space, processor power and battery life, making them weak, both in terms of security measures and computational capabilities. Hence, high-level security services, such as traditional cryptographic protocols that require key distributions or certificate management [2], might not be sufficiently efficient for IoT devices. Due to this, in recent years there has been a renewed effort into devising security schemes that reside in the physical layer and can supplement novel lightweight cryptographic protocols [3]–[5].

Physical-layer security constitutes a promising direction for securing the IoT. By using measurements of the common channel between them, two transceivers can agree to a bit sequence that can be then used as a seed for a cryptographic primitive or as an encryption key. Using the theory of reciprocity for antennas and electromagnetic propagation, and assuming that bidirectional transmissions occur inside the coherence time, methods [6], [7] have been proposed for the communicating parties to agree to a key based on these channel observations.

In this paper we present SKYGlow (Secret KeY Generator for LOW powered devices), a physical layer secret-key generation scheme designed for IoT devices. SKYGlow adopts a Discreet Cosine Transform (DCT) stage that can enhance the performance of key generation in comparison to the previously-proposed similar schemes. We evaluate the performance of SKYGlow on experimental data, collected in a test-bed of three off-the-shelf IEEE 802.15.4 transceivers that use the Texas Instrument's CC2650 chipset [8]. In line with previous works, we evaluate the entropy of the generated keys. In addition, we examine the likelihood of an eavesdropper managing to reconstruct the secret-key, assuming that she has access to all unencrypted information exchanged between the legitimate communicating parties.

The rest of the paper is organized as follows: Section II covers prior work in the domain. Section III defines the threat model and the characteristics of the eavesdropper, while Section IV describes our scheme, and elaborates on each stage. Section V describes our methodology and experimental set-up, and presents our results, followed by Section VI where we present our conclusions.

II. Prior work

The theoretical framework on secret-key extraction was laid in [9] where the authors examined the process of generating a common random key at two terminals, without letting an eavesdropper obtain information about this key. Since then, there have been a number of papers that present algorithms or implementations that can realise such a process with mixed results [10]–[14]. The process is generally as follows:

1) The legitimate communicating parties, Alice and Bob, exchange and observe messages transmitted in a potentially insecure manner.

2) The effects of the channel on the exchanged message are measured. This corresponds to the first process seen in Figure 1, and in our case it is a measurement of the Received Signal Strength (RSS) of each message.
3) The time-series created from the RSS values are quantized to result to a sequence of zeros and ones.
4) As the transceivers are not fully duplex there is a certain time delay between receiving and transmitting a message. This affects the reciprocity of the channel as it variates with time. Furthermore, our transceivers' RSS measurements can be imprecise. Thus the sequences of zeros and ones that Alice and Bob create might have some discrepancies. These are reconciled, either through some form of error correction [5], [15] or through some reconciliation protocol like *Cascade* [12]. There are also some cases where the scheme sacrifices entropy and randomness to result in two sequences that are completely similar after quantization, as seen in [16].
5) Finally, when the two sequences are identical, they typically go through the process of privacy amplification. This is necessary, as it is common for the bit sequences to have low entropy. By performing a transformation to increase key entropy and obfuscate any partial information an eavesdropper may have gathered during key reconciliation. However, this reduces the size of the secret-key.

SKYGlow works in a similar way, yet includes an extra stage between sampling the channel and quantization, that performs a DCT. As described in Section IV, all packet exchanges lead to at least one secret bit being generated, in contrast to other schemes in literature where a significant part of the RSS values of exchanged packets are unused. This greatly reduces the listening time and the number of wireless transmissions that need to take place, making the process significantly more energy-efficient. Furthermore the DCT stage is efficient and easily implementable in IEEE 802.15.4 hardware, making it ideal for IoT devices. Finally, our scheme generates keys with very high entropy (more than 0.9961 bits) making the Privacy Amplification stage unnecessary. This results in fewer processing cycles on each device as well as a larger secret-key which again makes our scheme more efficient.

III. Threat Model

The objective of a secret-key creation scheme is to extract a random, and shared by the two communicating parties, bit sequence, while preventing the eavesdropper from being able to reconstruct that key.

Before we continue then, we need to define the eavesdropper's abilities. We assume that the eavesdropper, Eve, can listen to all communications between the legitimate communicating parties, Alice and Bob. We also assume that the eavesdropper can record the RSS values of the overheard messages, although the resulting values are different than the ones Alice and Bob measure, as Eve is further away from both of them by at least the coherence distance [17]. Moreover, we assume that Eve is completely passive, and she does not attempt to jam the medium, inject traffic or in general transmit at any time. We make no assumption on the hardware capabilities of Eve. We assume Eve has access to the syndrome transmitted from Alice to Bob for error correction, and knows how the protocol works, thus can try to use the syndrome to recover the secret-key. For the rest of this paper the terms Alice, Bob and Eve are used interchangeably with A, B and E respectively.

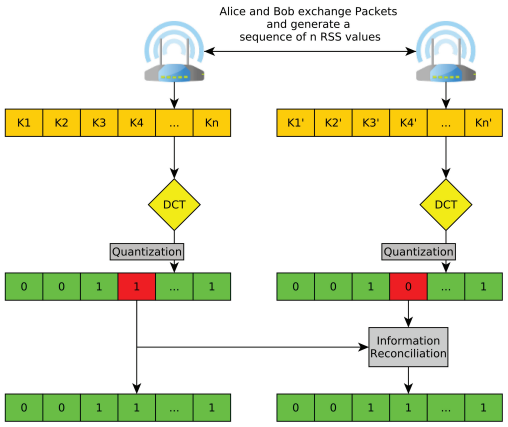

Fig. 1. Overview of SKYGlow.

IV. Proposed Scheme

In this section we elaborate on each stage of SKYGlow and highlight how our scheme innovates compared to other schemes in the existing literature. We stress that SKYGlow does not require a privacy amplification stage since the entropy of the generated keys is sufficiently high without it.

A. Sampling and DCT

Since SKYGlow is targeted for IoT devices with simple hardware, *i.e.* one antenna and low computationally capabilities, we use the RSS values of the received packets as the sampling method on the effects of the channel on the exchanged messages. We should note, that we have no information about how exactly the RSS of each packet is calculated. We proceed assuming that Alice and Bob calculate the RSS values in the same way.

Instead of using the RSS values as input to the quantizer, we first apply a DCT to those values. The DCT expresses a finite sequence of data points (our RSS values) in terms of a sum of cosine functions oscillating at different frequencies, according to the following equation:

$$X_k = \sum_{n=0}^{N-1} x_n cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, ..., N \qquad (1)$$

The major theoretical advantage of including a DCT stage is that it allows us to discard at will the higher frequency components that are mostly responsible for the bit mismatches. Thus, the scheme is, in a sense tunable, and able to generate more secret bits when there is a high degree of reciprocity, or fewer bits when the channel is less symmetrical (due to

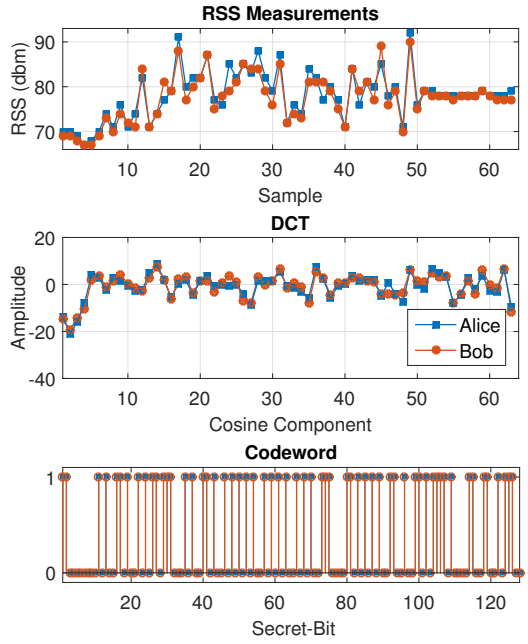

Fig. 2. Top: An example of 64 RSS measurements. Middle: The respective DCT transformation. Bottom: The final 128-bit secret key.

communications being half-duplex or if the coherence time of the channel is less than the sampling period). An illustration of the results of applying the DCT to a sequence of RSS values can be seen in Figure 2.

B. Quantization

We then quantize x_n and use the quantized values to produce the secret-key. The more symmetrical the channel is, the more cosine components will have the same amplitude.

A significant fraction of secret-key extraction schemes in the literature use a quantizer like the one presented in [16], or a variation of it. In the aforementioned, the mean, μ and the standard deviation σ of the measurements are first calculated. By using a coefficient $\alpha > 0$, all the measurements that are in the region $[\mu + \alpha\sigma, \mu - \alpha\sigma]$ are discarded, in effect creating a censoring region. Every value that is larger than $\mu + \alpha\sigma$ is then quantized as a 1 and everything smaller than $\mu - \alpha\sigma$ is quantized as 0. That ensures that only the high and low extremes of the measurements, that have a higher degree of correlation and are not a result of thermal noise, are actually used for the key generation process. The coefficient α is used to optimize the size of the censoring region. This approach has two disadvantages. First, when Alice logs a value that is outside her censoring region, but Bob's measurement is just inside it, the two sequences are desynchronized, which leads to further errors. Moreover, the censoring region leads to measurements being discarded, which translates to energy being wasted. In the energy-conscious elements of the IoT such a process is very inefficient.

Our proposed quantizer retains the use of μ and σ, however these are calculated from the amplitudes of the cosine waves after the DCT transform. In our case there is no censoring region, instead the quantizer works with the following simple formula:

$$Q(n) = \begin{cases} 11, & F(x) \geq \mu + \sigma, \\ 10, & F(x) \in [\mu, \mu + \sigma), \\ 01, & F(x) \in (\mu - \sigma, \mu), \\ 00, & F(x) \leq \mu - \sigma. \end{cases}$$

C. Information Reconciliation

Due to communications being half-duplex, the channel not being completely symmetrical and the effects of thermal noise, the RSS values between Alice and Bob have small deviations. These small deviations can lead to deviations in the amplitude of the cosine waves after the DCT and eventually, after quantization occurs, to bit mismatches.

Other works in this field employ a variety of methods to deal with these mismatches. In [16] the RSS values are filtered, to discard the high-frequency components that are not correlated, but that significantly reduces the entropy of the sequence. Another common practice in related schemes is to employ the Cascade Protocol [18]. However Cascade requires a large number of channel communications between the parties to proceed, which is unsuitable for IoT applications where the devices are often battery-powered and communication comes at a great energy cost [19].

SKYGlow can discard the cosine waves of higher frequencies that tend to be responsible for the mismatches, depending on the requirements of the implementation environment. That can reduce the errors, but it also results in fewer secret-bits per packet. Although that could be acceptable for certain applications, in this paper, we opt for the approach that would generate an 128-bit secret-key with the fewer number of transmissions. Thus, we implement Information Reconciliation with Slepian-Wolf Low Density Parity Codes (LDPC). To reduce the amount of wireless transmissions needed to correct mismatches, we limited the syndrome length to the largest that would fit in the link layer payload of a IEEE 802.15.4 frame, that is 1016 bits. Thus for error correction to occur we need to send only one packet after the RSS values have been collected. To satisfy that requirement for codewords of 128-bit length, we employed a code rate of 7.

V. Experimental Evaluation

A. Implementation

We implemented our system on three SmartRF06 evaluation boards [20] using the CC2650 radio [8]. Two of the boards act as Alice and Bob, while the third acts as Eve. In Figure 4, we detail the spatial arrangement of Alice, Bob and Eve in each scenario.

Commercial wireless transceivers are currently half-duplex, thus the logged RSS values correspond to messages that have a small time delay of each other. The process of message exchange can be seen in Figure 3, where t_p is the transmission and propagation delay, t_c the time needed for Bob to measure the RSS value of the message and respond with another message, and t_f the time between two successive messages sent

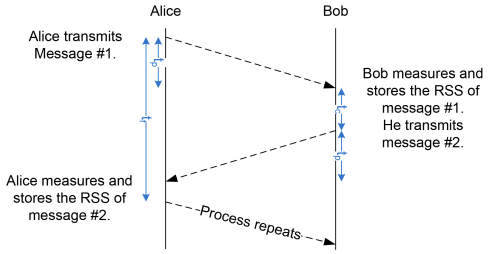

Fig. 3. The methodology of RSS logging.

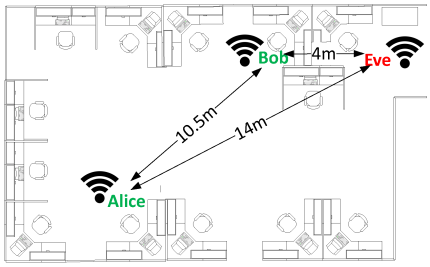

Fig. 4. Layout of experiment in office space.

by Alice. To ensure the reciprocity of radio wave propagation we have to keep the time between Alice's first transmission and Alice's reception of a response as small as possible, and lower than the coherence time of the channel. Successive transmissions from Alice should take place after the coherence time has elapsed though, otherwise the successive RSS values are correlated and thus less random, which reduces the entropy of the generated key. For the purposes of this work $t_p = 2.4$ ms while $t_c = 7.8$ ms and $t_f = 1$ s.

Although SKYGlow can be tuned to produce secret-keys of different size, we are mainly interested in IoT applications. Hence, we aim to generate an 128-bit secret-key, with as few transmissions as possible while keeping the entropy of the keys high. As the quantizer of SKYGlow described in Section IV-B, produces 2 bits for every DCT wave component, we use sets of 64 RSS values and a LDPC code of rate 7. Thus, for generating an 128-bit secret-key, the scheme requires the exchange of 65 packets, including the syndrome, meaning that up to 1.96 secret bits per packet are generated.

B. Evaluation Metrics

To assess the performance of SKYGlow, we employ the following metrics:

- Bit Error Rate (BER): similar to the way BER is used in communication systems, we use BER to denote the ratio of mismatches between the key that Alice and Bob extract. Ideally BER would be 0 for Alice and Bob, and 0.5 for Eve.
- Key Agreement Rate (KAR): the probability of agreeing on a key each time the scheme is executed. Ideally KAR is equal to 1.
- Key Leakage Rate (KLR): the probability that Eve reconstructs the correct secret-key from her observations and by using the insecurely transmitted syndrome. Ideally KLR is equal to 0.
- Secret Bits per Packet (SBP): the number of secret bits generated from each packet exchange between Alice and Bob. The highest the SBP, the fewer transmissions are required for a key generation; thus, the more energy-efficient the system is.

In addition, it is possible that the BER is so high that the LDPC decoding is unable to correct the errors. Assuming that KAR is denoted as $1-p$, and that it takes N packets to generate a key, then the long term average number of packets needed to agree to a key is:

$$E[N] = \sum_{n=1}^{\infty} Np^{n-1}(1-p)n = \frac{N}{1-p},$$

where $N = 65$ in our implementation.

Let us envision a security protocol that executes SKYGlow periodically to generate new secret-keys with every block of N packets. In this scenario, only the first syndrome would be communicated unencrypted, while the following syndromes can be encrypted with the previously generated key. Assuming that KLR is denoted as q, we can estimate the long-term average number of leaked packets, M, as follows:

$$E[M] = \sum_{m=1}^{\infty} q^m \cdot E[N] = \frac{q}{1-q}\frac{N}{1-p}.$$

$E[N]$ and $E[M]$ are also used as performance metrics, considering that a smaller $E[N]$ results to a more efficient system, while a smaller $E[M]$ suggests a more secure system.

C. Experimental Results

We evaluate SKYGlow in two realistic scenarios: an office space with stationary terminals, and an office space with a link between a mobile terminal and a stationary terminal. In these scenarios, we collected sets of RSS values that lasted from several hours (for the mobile scenarios) up to several days (for the stationary scenarios). The results are summarized in Table I.

1) Scenario 1: Office Space with stationary terminals: We first evaluate SKYGlow in an open office space whose layout can be seen in Figure 4. This scenario represents a very likely application, especially as IoT devices find themselves in residential and commercial locations. We examine two different cases:

- The case where Alice and Bob have Line of Sight (LoS), both in working and non-working hours
- The case where Alice and Bob do not have a direct Line of Sight (nLos), but have to rely on multipaths for communication. Again we examine both working and non-working hours.

The LOS case is characterized by a strong dominant component. Furthermore, during working hours, defined as the period between 09:00-17:00, people are working in their offices with

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS

	Stationary		Mobile
	LoS	NLoS	
A's and B's BER	0.0323	0.0072	0.0351
E's BER	0.4832	0.4830	0.4853
KAR	0.8348	0.8970	0.8438
KLR	0.0067	0.01	0.0042
SBP	1.64	1.77	1.66
Average Key Entropy	0.9971	0.9977	0.9969
E[N]	77.66	72.34	76.91
E[M]	0.52	0.73	0.28

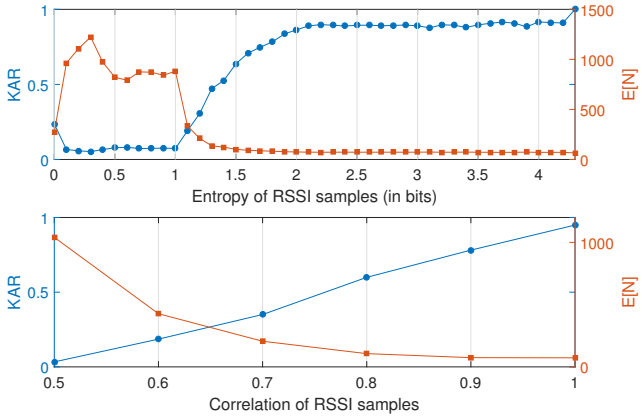

Fig. 5. Top: KAR (blue) and E[N] (red) as a function of entropy. Bottom: KAR (blue) and E[N] (red) as a function of correlation.

frequent movement. This leads to increased entropy [21] and deviation of the RSS measurements. The KAR is 0.8348 and the long term average of packets required to agree to an 128=bit key is 77.66, meaning that the SBP is 1.64 on average. When lacking a direct line of sight between Alice and Bob, communication relies on the multipaths, thus there is a slightly higher entropy than in the LoS case. The KAR is 0.8970 and the long term average of packets required to agree to an 128 bit-key is 72.34, meaning that the SBP is 1.77 on average.

2) Scenario 2: Mobile terminals: In the second scenario we examine the communication between a stationary base-station and a mobile terminal. This scenario was run in two different circumstances, with the results being similar. As Alice is moving around and out of the office space, LoS is lost and re-established, the pathways change and thus there was greater variation in the logged RSS values. The KAR is 0.8438 and the long term average of packets required to agree to an 128-bit key is 76.91, meaning that the SBP is 1.66 on average.

D. Discussion on the Results

From the above, it is easy to see that although there is a chance of not agreeing on the key on the first attempt, the probability of not agreeing on the second attempt is only 0.024, while on the third attempt is 0.0038. Thus it is virtually guaranteed that after 195 packets a secret key is extracted.

Indeed, when a key generation fails, we recommend a new attempt with a fresh block of 65 packets. An alternative approach could be followed instead. When a disagreement occurs, instead of discarding all packets and starting over, Alice and Bob, could just use a quantizer that would result in only one bit per DCT component, or define a censoring region and use the more half more extreme measurements, who would be better correlated. Although that increases the chance of success, it follows that from our 64 samples, we would only end with 64 bits, necessitating another 64 packets to be transmitted. Thus we end up with 130 packets transmitted (128 for the RSS values and 2 syndromes), which is exactly the same as repeating the process once more.

The entropy and the correlation of the RSS values are, in fact, two of the most important factors that govern the performance of SKYGlow. Indeed, when the entropy is low, like for example when the environment is almost completely static, the changes in the RSS values are often due to thermal noise or the result of a cause that is short enough to affect only one side of the exchange. This creates uncorrelated randomness that negatively affects performance. This is presented more clearly in Figure 5. It can be observed that when the correlation becomes less than 0.7 the long-term average of packets needed to agree to a key becomes larger than 200. Similarly, when the average entropy of the RSS value falls below 1.2 bits, the long-term average required messages is more than 210.

However, it is important to note that for the vast majority of the time during working hours, correlation is over 0.8 and entropy over 1.5 bits for both the stationary scenario as well as the mobile one, as can be seen in the cumulative distribution graphs in Figure 6. Yet, during non-working hours, when the environment is almost static and RSS variation is a result of thermal noise and thus uncorrelated and unshared, performance degenerates as entropy and correlation both reduce.

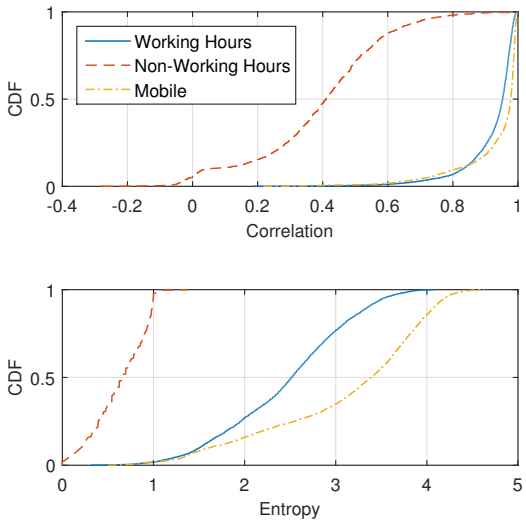

Fig. 6. Empirical cumulative distribution as a function of the correlation of our samples (top) and the entropy of our sequences (bottom)

TABLE II
PERFORMANCE COMPARISON

	SBP	Key Entropy
Patwari et al. [13]	0.05 – 0.44	0.9590
Ali et al. [16]	0.33 – 0.0370	0.9979
Revadigar et al. [23]	1	N/A
SKYGlow	1.64 – 1.7	0.9971

E. Comparison with Other Schemes

In Table II, we present a brief comparison of the performance of SKYGlow against several state-of-the-art schemes designed for IoT devices. We note that some of this works use as a metric the number of secret-bits generated per second. We consider that an inadequate metric for energy-constraint IoT scenarios that are characterised by sparse communications. Thus, we use the SBP metric that reflects the efficiency of the key generation process. Furthermore, we limit our comparison to schemes implemented on wither IEEE 802.15.4 or BLE hardware. It can be observed that, in line with previous works, SKYGlow generates keys with a very high entropy. Yet the keys are generated in a more efficient manner, requiring fewer transmissions between Alice and Bob. From the table we omit [22], one of the newest proposed schemes, as from the published paper there is no way to derive its SBP. We note however that it produces up to 0.195 bits/second, while SKYGlow produces more than 1.64 bits/second.

VI. CONCLUSIONS

In this paper, we present a novel method for secret-key generation between two IoT transceivers, with an emphasis on energy conservation. Our contribution, named SKYGlow, incorporates a DCT stage on the raw RSS samples, resulting in better performance and sufficient entropy that makes a privacy amplification stage unnecessary. Transforming the RSS sequence with the DCT allows us to more finely tune the scheme, as uncorrelated higher-frequency components can be discarded at will. Finally Slepian Wolf Coding is employed for information reconciliation. We verify our results on experimental data collected with off-the-shelf hardware, and demonstrate that SKYGlow can generate 1.66 secret bits per packet for mobile scenarios and 1.77 secret-bits per packet for stationary scenarios, outperforming the current state-of-the-art schemes for IoT devices, and allowing for both faster and more efficient secret-key generation.

ACKNOWLEDGEMENT

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) through grants EP/I028153/1 and EP/K031910/1 (IRC-SPHERE), and the University of Bristol. Many thanks to Giannis Moutsinas for the inspiring discussions.

REFERENCES

[1] F. Mattern and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," in *From active data management to event-based systems and more*. Springer, 2010, pp. 242–259.

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[3] A. F. Skarmeta, J. L. Hernandez-Ramos, and M. Moreno, "A decentralized approach for security and privacy challenges in the Internet of Things," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 67–72.

[4] M. Abomhara and G. M. Koien, "Security and privacy in the Internet of Things: Current status and open issues," in *Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on*. IEEE, 2014, pp. 1–8.

[5] C. Ye and P. Narayan, "Secret key and private key constructions for simple multiterminal source models," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 639–651, 2012.

[6] J. Muramatsu, K. Yoshimura, P. Davis, A. Uchida, and T. Harayama, "Secret-Key Distribution Based on Bounded Observability," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1762–1780, 2015.

[7] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *Digital Signal Processing*, vol. 6, no. 4, pp. 207–212, 1996.

[8] Texas Instruments, "CC2650 SimpleLink Multistandard Wireless MCU," http://www.ti.com/lit/ds/symlink/cc2650.pdf, 2015.

[9] R. Ahlswede and I. CSISEAR, "Common randomness in information theory and cryptography. i: Secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

[10] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 128–139.

[11] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on mobile Computing*, vol. 12, no. 5, pp. 917–930, 2013.

[12] S. N. Premnath, J. Croft, N. Patwari, and S. K. Kasera, "Efficient high-rate secret key extraction in wireless sensor networks using collaboration," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, p. 2, 2014.

[13] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.

[14] A. J. Pierrot, R. A. Chou, and M. R. Bloch, "Experimental aspects of secret key generation in indoor wireless environments," in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2013, pp. 669–673.

[15] M. Bloch, J. Barros, M. R. Rodrigues, and S. W. McLaughlin, "Wireless information-theoretic security," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2515–2534, 2008.

[16] S. T. Ali, V. Sivaraman, and D. Ostry, "Zero reconciliation secret key generation for body-worn health monitoring devices," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 2012, pp. 39–50.

[17] G. D. Durgin and T. S. Rappaport, "Theory of multipath shape factors for small-scale fading wireless channels," *Antennas and Propagation, IEEE Transactions on*, vol. 48, no. 5, pp. 682–693, 2000.

[18] G. Brassard and L. Salvail, "Secret-Key Reconciliation by Public Discussion." Springer-Verlag, 1994, pp. 410–423.

[19] J. Martinez-Mateo, C. Pacher, M. Peev, A. Ciurana, and V. Martin, "Demystifying the information reconciliation protocol cascade," *arXiv preprint arXiv:1407.3257*, 2014.

[20] Texas Instruments, "SmartRF06 Evaluation Board Users Guide," http://www.ti.com/lit/ug/swru321a/swru321a.pdf, 2013.

[21] G. Margelis, X. Fafoutis, G. Oikonomou, R. J. Piechocki, T. Tryfonas, and P. Thomas, "Practical limits of the secret Key-Capacity for IoT physical layer security," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, USA, Dec. 2016.

[22] S. A. Salehi, M. Razzaque, I. Tomeo-Reyes, N. Hussain, and V. Kaviani, "Efficient high-rate key management technique for wireless body area networks," in *Communications (APCC), 2016 22nd Asia-Pacific Conference on*. IEEE, 2016, pp. 529–534.

[23] G. Revadigar, C. Javali, W. Xu, W. Hu, and S. Jha, "Secure key generation and distribution protocol for wearable devices," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2016, pp. 1–4.